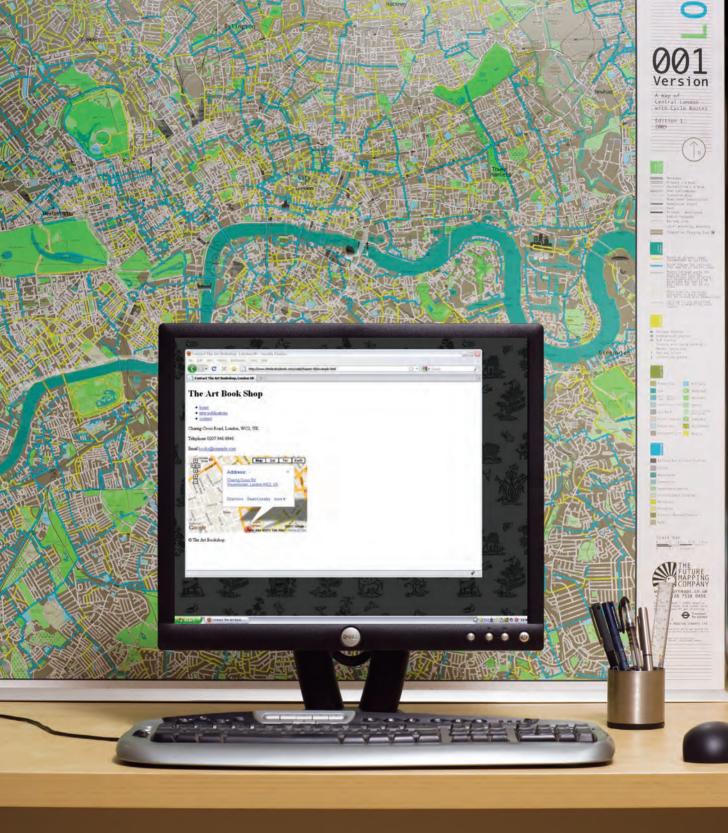# 8

EXTRA MARKUP

- ▸ Specifying different versions of HTML
- ▸ Identifying and grouping elements
- ▸ Comments, meta information and iframes

At this point, we have covered the main tags that fit nicely into groups and sections.

In this chapter, we will focus on some helpful topics that are not easily grouped together. You will learn about:

- The different versions of HTML and how to indicate which version you are using

- How to add comments to your code

- Global attributes, which are attributes that can be used on any element, including the `class` and `id` attributes

- Elements that are used to group together parts of the page where no other element is suitable

- How to embed a page within a page using iframes

- How to add information about the web page using the `<meta>` element

- Adding characters such as angled brackets and copyright symbols

THE EVOLUTION OF HTML

Since the web was first created, there have been several different versions of HTML.

HTML 4
RELEASED 1997

XHTML 1.0
RELEASED 2000

Each new version was designed to be an improvement on the last (with new elements and attributes added and older code removed).

There have also been several versions of each browser used to view web pages, each of which implements new code. Not all web users, however, have the latest browsers installed on their computers, which means that not everyone will be able to view all of the latest features and markup.

Where you should be particularly aware of browsers not supporting certain features, I have made a note of this (as you have seen with some of the HTML5 elements introduced in the Forms chapter — and as you will see in the CSS chapters).

With the exception of a few elements added in HTML5 (which have been highlighted), the elements you have seen in this book were all available in HTML 4.

Although HTML 4 had some presentational elements to control the appearance of pages, authors are not recommended to use them any more. (Examples include the `<center>` element for centering content on a page, `<font>` for controlling the appearance of text, and `<strike>` to put a line through the text — all of these can be achieved with CSS instead.)

In 1998, a language called XML was published. Its purpose was to allow people to write new markup languages. Since HTML was the most widely used markup language around, it was decided that HTML 4 should be reformulated to follow the rules of XML and it was renamed XHTML. This meant that authors had to follow some new, more strict rules about writing markup. For example:

- Every element needed a closing tag (except for empty elements such as `<img />`).
- Attribute names had to be in lowercase.
- All attributes required a value, and all values were to be placed in double quotes.
- Deprecated elements should no longer be used.
- Every element that was opened inside another element should be closed inside that same element.

The examples in this book all follow these strict rules of XML.

One of the key benefits of this change was that XHTML works seamlessly with other programs that are written to create and process XML documents.

It could also be used with other data formats such as Scalable Vector Graphics (SVG) — a graphical language written in XML, MathML (used to mark up mathematical formulae), and CML (used to mark up chemical formulae).

In order to help web page authors move to this new syntax, two main flavors of XHTML 1.0 were created:

- **Strict XHTML 1.0**, where authors had to follow the rules to the letter
- **Transitional XHTML 1.0**, where authors could still use presentational elements (such as `<center>` and `<font>`).

The transitional version of XHTML was created because it allowed authors to continue to follow older practices (with a less strict syntax) and use some of the elements and attributes that were going to be removed from future versions of HTML.

There was also a third version of XHTML 1.0 called **XHTML 1.0 Frameset**, which allowed web page authors to partition a browser window into several "frames," each of which would hold a different HTML page. These days, frames are very rarely used and are being phased out.

HTML5
RELEASED 2000

In HTML5, web page authors do not need to close all tags, and new elements and attributes will be introduced. At the time of writing, the HTML5 specification had not been completed, but the major browser makers had started to implement many of the new features, and web page authors were rapidly adopting the new markup.

Despite the fact that HTML5 is not yet completed, you can safely take advantage of the new features of the language as long as you endeavour to ensure that users with older browsers will be able to view your pages (even though some of the extra features will not be visible to them).

DOCTYPES

Because there have been several versions of HTML, each web page should begin with a DOCTYPE declaration to tell a browser which version of HTML the page is using (although browsers usually display the page even if it is not included). We will therefore be including one in each example for the rest of the book.

As you will see when we come to look at CSS and its box model on page 316, the use of a DOCTYPE can also help the browser to render a page correctly.

Because XHTML was written in XML, you will sometimes see pages that use the XHTML strict DOCTYPE start with the optional XML declaration. Where this is used, it should be the first thing in a document. There must be nothing before it, not even a space.

HTML5   **HTML**

```
<!DOCTYPE html>
```

HTML 4

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
```

Transitional XHTML 1.0

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/
   xhtml1-transitional.dtd">
```

Strict XHTML 1.0

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/
   xhtml1-strict.dtd">
```

XML Declaration

```
<?xml version="1.0" ?>
```

COMMENTS IN HTML

```
<!-- start of introduction -->
<h1>Current Exhibitions</h1>
<h2>Olafur Eliasson</h2>
<!-- end of introduction -->
<!-- start of main text -->
<p>Olafur Eliasson was born in Copenhagen, Denmark
   in 1967 to Icelandic parents.</p>
<p>He is known for sculptures and large-scale
   installation art employing elemental materials
   such as light, water, and air temperature to
   enhance the viewer's experience.</p>
<!-- end of main text -->
<!--
  <a href="mailto:info@example.org">Contact</a>
-->
```

RESULT

Current Exhibitions

Olafur Eliasson

Olafur Eliasson was born in Copenhagen, Denmark in 1967 to Icelandic parents.

He is known for sculptures and large-scale installation art employing elemental materials such as light, water, and air temperature to enhance the viewer's experience.

<!-- -->

If you want to add a comment to your code that will not be visible in the user's browser, you can add the text between these characters:

`<!-- comment goes here -->`

It is a good idea to add comments to your code because, no matter how familiar you are with the page at the time of writing it, when you come back to it later (or if someone else needs to look at the code), comments will make it much easier to understand.

Although comments are not visible to users in the main browser window, they can be viewed by anyone who looks at the source code behind the page.

On a long page you will often see comments used to indicate where sections of the page start or end, and to pass on notes to help anyone who is looking at the code understand it.

Comments can also be used around blocks of code to stop that code from being displayed in the browser. In the example on the left, the email link has been commented out.

ID ATTRIBUTE

Every HTML element can carry the `id` attribute. It is used to uniquely identify that element from other elements on the page. Its value should start with a letter or an underscore (not a number or any other character). It is important that no two elements on the same page have the same value for their `id` attributes (otherwise the value is no longer unique).

As you will see when you come to look at CSS in the next section, giving an element a unique identity allows you to style it differently than any other instance of the same element on the page. For example, you might want to assign one paragraph within the page (perhaps a paragraph containing a pull quote) a different style than all of the other paragraphs. In the example on the right, the paragraph with the `id` attribute whose value is `pullquote` is made uppercase using CSS.

If you go on to learn about JavaScript (a language that allows you to add interactivity to your pages), `id` attributes can be used to allow the script to work with that particular element.

The `id` attribute is known as a **global attribute** because it can be used on any element.

```
chapter-08/id-attribute.html                    HTML
<p>Water and air. So very commonplace are these
   substances, they hardly attract attention - and
   yet they vouchsafe our very existence.</p>
<p id="pullquote">Every time I view the sea I feel
   a calming sense of security, as if visiting my
   ancestral home; I embark on a voyage of seeing.
   </p>
<p>Mystery of mysteries, water and air are right
   there before us in the sea.</p>
```

RESULT

Water and air. So very commonplace are these substances, they hardly attract attention - and yet they vouchsafe our very existence.

EVERY TIME I VIEW THE SEA I FEEL A CALMING SENSE OF SECURITY, AS IF VISITING MY ANCESTRAL HOME; I EMBARK ON A VOYAGE OF SEEING.

Mystery of mysteries, water and air are right there before us in the sea.

CLASS ATTRIBUTE

```
<p class="important">For a one-year period from
    November 2010, the Marugame Genichiro-Inokuma
    Museum of Contemporary Art (MIMOCA) will host a
    cycle of four Hiroshi Sugimoto exhibitions.</p>
<p>Each will showcase works by the artist
    thematically contextualized under the headings
    "Science," "Architecture," "History" and
    "Religion" so as to present a comprehensive
    panorama of the artist's oeuvre.</p>
<p class="important admittance">Hours: 10:00 - 18:00
    (No admittance after 17:30)</p>
```

RESULT

FOR A ONE-YEAR PERIOD FROM NOVEMBER 2010,
THE MARUGAME GENICHIRO-INOKUMA MUSEUM
OF CONTEMPORARY ART (MIMOCA) WILL HOST A
CYCLE OF FOUR HIROSHI SUGIMOTO EXHIBITIONS.

Each will showcase works by the artist thematically
contextualized under the headings "Science," "Architecture,"
"History" and "Religion" so as to present a comprehensive
panorama of the artist's oeuvre.

HOURS: 10:00 – 18:00 (NO ADMITTANCE AFTER 17:30)

Every HTML element can also carry a class attribute. Sometimes, rather than uniquely identifying one element within a document, you will want a way to identify several elements as being different from the other elements on the page. For example, you might have some paragraphs of text that contain information that is more important than others and want to distinguish these elements, or you might want to differentiate between links that point to other pages on your own site and links that point to external sites.

To do this you can use the class attribute. Its value should describe the class it belongs to. In the example on the left, key paragraphs have a class attribute whose value is important.
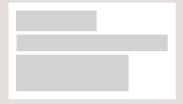
The class attribute on any element can share the same value. So, in this example, the value of important could be used on headings and links, too.

By default, using these attributes does not affect the presentation of an element. It will only change their appearance if there is a CSS rule that indicates it should be displayed differently.

In this example, CSS has been applied to make elements with a class attribute whose value is important uppercase, and elements with a class attribute whose value is admittance red.

If you would like to indicate that an element belongs to several classes, you can separate class names with a space, as you can see in the third paragraph in the example above.

BLOCK ELEMENTS

Some elements will always appear to start on a new line in the browser window. These are known as **block level** elements.

Examples of block elements are <h1>, <p>, <ul>, and <li>.

```html
<h1>Hiroshi Sugimoto</h1>
<p>The dates for the ORIGIN OF ART exhibition are as
   follows:</p>
<ul>
  <li>Science: 21 Nov - 20 Feb 2010/11</li>
  <li>Architecture: 6 Mar - 15 May 2011</li>
  <li>History: 29 May - 21 Aug 2011</li>
  <li>Religion: 28 Aug - 6 Nov 2011</li>
</ul>
```

RESULT

Hiroshi Sugimoto

The dates for the ORIGIN OF ART exhibition are as follows:

- Science: 21 Nov - 20 Feb 2010/11
- Architecture: 6 Mar - 15 May 2011
- History: 29 May - 21 Aug 2011
- Religion: 28 Aug - 6 Nov 2011

INLINE ELEMENTS

```
Timed to a single revolution of the planet around
the sun at a 23.4 degrees tilt that plays out the
rhythm of the seasons, this <em>Origins of Art</em>
cycle is organized around four themes: <b>science,
architecture, history</b> and <b>religion</b>.
```

RESULT

Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this *Origins of Art* cycle is organized around four themes: **science, architecture, history** and **religion**.

Some elements will always appear to continue on the same line as their neighbouring elements. These are known as **inline** elements.

Examples of inline elements are <a>, <b>, <em>, and <img>.

GROUPING TEXT & ELEMENTS IN A BLOCK

`<div>`

The `<div>` element allows you to group a set of elements together in one block-level box.
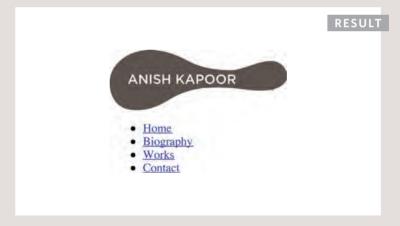
For example, you might create a `<div>` element to contain all of the elements for the header of your site (the logo and the navigation), or you might create a `<div>` element to contain comments from visitors.

In a browser, the contents of the `<div>` element will start on a new line, but other than this it will make no difference to the presentation of the page.

Using an `id` or `class` attribute on the `<div>` element, however, means that you can create CSS style rules to indicate how much space the `<div>` element should occupy on the screen and change the appearance of all the elements contained within it.

It can also make it easier to follow your code if you have used `<div>` elements to hold each section of the page.

HTML

```html
<div id="header">
  <img src="images/logo.gif" alt="Anish Kapoor" />
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="biography.html">Biography</a></li>
    <li><a href="works.html">Works</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</div><!-- end of header -->
```

RESULT

Since there may be several other elements inside a `<div>` element, it can be helpful to add a comment after the closing `</div>` tag.

This allows you to clearly see which opening tag it is supposed to correspond to, as shown at the end of the example here.

GROUPING TEXT & ELEMENTS INLINE

```html
<p>Anish Kapoor won the Turner Prize in 1991 and
   exhibited at the <span class="gallery">Tate
   Modern</span> gallery in London in 2003.</p>
```

RESULT

Anish Kapoor won the Turner Prize in 1991 and exhibited at the
TATE MODERN gallery in London in 2003.

`<span>`

The `<span>` element acts like an inline equivalent of the `<div>` element. It is used to either:

1. Contain a section of text where there is no other suitable element to differentiate it from its surrounding text

2. Contain a number of inline elements

The most common reason why people use `<span>` elements is so that they can control the appearance of the content of these elements using CSS.

You will usually see that a `class` or `id` attribute is used with `<span>` elements:

- To explain the purpose of this `<span>` element

- So that CSS styles can be applied to elements that have specific values for these attributes

IFRAMES

`<iframe>`

An iframe is like a little window that has been cut into your page — and in that window you can see another page. The term iframe is an abbreviation of inline frame.

One common use of iframes (that you may have seen on various websites) is to embed a Google Map into a page. The content of the iframe can be any html page (either located on the same server or anywhere else on the web).

An iframe is created using the `<iframe>` element. There are a few attributes that you will need to know to use it:

src
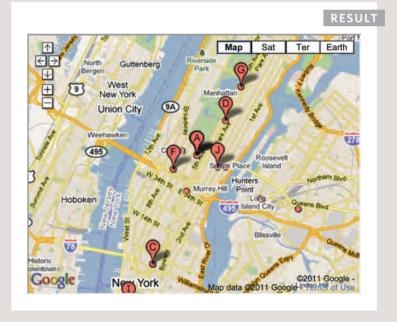The `src` attribute specifies the URL of the page to show in the frame.

height
The `height` attribute specifies the height of the iframe in pixels.

width
The `width` attribute specifies the width of the iframe in pixels.

chapter-08/iframes.html   HTML

```
<iframe
  width="450"
  height="350"
  src="http://maps.google.co.uk/maps?q=moma+new+york
  &amp;output=embed">
</iframe>
```

RESULT

```
<iframe
  src="http://maps.google.co.uk/maps?q=moma+new+york
  &amp;output=embed"
  width="450"
  height="350"
  frameborder="0"
  scrolling="no">
</iframe>
```

RESULT

scrolling

The scrolling attribute will not be supported in HTML5. In HTML 4 and XHTML, it indicates whether the iframe should have scrollbars or not. This is important if the page inside the iframe is larger than the space you have allowed for it (using the height and width attributes). Scrollbars allow the user to move around the frame to see more content. It can take one of three values: yes (to show scrollbars), no (to hide scrollbars) and auto (to show them only if needed).

frameborder

The frameborder attribute will not be supported in HTML5. In HTML 4 and XHTML, it indicates whether the frame should have a border or not. A value of 0 indicates that no border should be shown. A value of 1 indicates that a border should be shown.

seamless

In HTML5, a new attribute called seamless can be applied to an iframe where scrollbars are not desired. The seamless attribute (like some other new HTML5 attributes) does not need a value, but you will often see authors give it a value of seamless. Older browsers do not support the seamless attribute.

INFORMATION ABOUT YOUR PAGES

`<meta>`

The `<meta>` element lives inside the `<head>` element and contains information about that web page.

It is not visible to users but fulfills a number of purposes such as telling search engines about your page, who created it, and whether or not it is time sensitive. (If the page is time sensitive, it can be set to expire.)

The `<meta>` element is an empty element so it does not have a closing tag. It uses attributes to carry the information.

The most common attributes are the `name` and `content` attributes, which tend to be used together. These attributes specify properties of the entire page. The value of the `name` attribute is the property you are setting, and the value of the `content` attribute is the value that you want to give to this property.

In the first line of the example on the opposite page, you can see a `<meta>` element where the `name` attribute indicates an intention to specify a description for the page. The `content` attribute is where this description is actually specified.

The value of the `name` attribute can be anything you want it to be. Some defined values for this attribute that are commonly used are:

description

This contains a description of the page. This description is commonly used by search engines to understand what the page is about and should be a maximum of 155 characters. Sometimes it is also displayed in search engine results.

keywords

This contains a list of comma-separated words that a user might search on to find the page. In practice, this no longer has any noticeable effect on how search engines index your site.

robots

This indicates whether search engines should add this page to their search results or not. A value of `noindex` can be used if this page should not be added. A value of `nofollow` can be used if search engines should add this page in their results but not any pages that it links to.

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Information About Your Pages</title>
    <meta name="description"
      content="An Essay on Installation Art" />
    <meta name="keywords"
      content="installation, art, opinion" />
    <meta name="robots"
      content="nofollow" />
    <meta http-equiv="author"
      content="Jon Duckett" />
    <meta http-equiv="pragma"
      content="no-cache" />
    <meta http-equiv="expires"
      content="Fri, 04 Apr 2014 23:59:59 GMT" />
  </head>
  <body>
  </body>
</html>
```

The <meta> element also uses the http-equiv and content attributes in pairs. In our example, you can see three instances of the http-equiv attribute. Each one has a different purpose:

author

This defines the author of the web page.

pragma

This prevents the browser from caching the page. (That is, storing it locally to save time downloading it on subsequent visits.)

expires

Because browsers often cache the content of a page, the expires option can be used to indicate when the page should expire (and no longer be cached). Note that the date must be specified in the format shown.

ESCAPE CHARACTERS

There are some characters that are used in and reserved by HTML code. (For example, the left and right angled brackets.)

Therefore, if you want these characters to appear on your page you need to use what are termed "escape" characters (also known as escape codes or entity references). For example, to write a left angled bracket, you can use either `<` or `<`. For an ampersand, you can use either `&` or `&`.

There are also special codes that can be used to show symbols such as copyright and trademark, currency symbols, mathematical characters, and some punctuation marks. For example, if you want to include a copyright symbol on a web page you can use either `©` or `©`.

When using escape characters, it is important to check the page in your browser to ensure that the correct symbol shows up. This is because some fonts do not support all of these characters and you might therefore need to specify a different font for these characters in your CSS code.

ONLINE EXTRA
You can find a more complete list of escape codes in the tools section of the website accompanying this book.

< Less-than sign
<
<

> Greater-than sign
>
&

& Ampersand
&
&

" Quotation mark
"
"

¢ Cent sign
¢
¢

£ Pound sign
£
£

¥ Yen sign
¥
¥

€ Euro sign
€
€

© Copyright symbol
©
©

® Registered trademark
®
®

™ Trademark
™
™

' Left single quote
‘
‘

' Right single quote
’
’

" Left double quotes
“
“

" Right double quotes
”
”

× Multiplication sign
×
×

÷ Division sign
÷
÷

This example starts by using a DOCTYPE to indicate that this is an HTML 4 web page. In the head, you can also see a <meta> tag describing the page's content. Several elements use the id and class attributes to identify their purpose. The copyright symbol has been added using an escape code.

Parts of the page have been grouped using <div> elements, and comments have been added to indicate what the </div> elements are closing.

EXAMPLE
EXTRA MARKUP

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta name="description" content="Telephone, email
      and directions for The Art Bookshop, London, UK" />
    <title>Contact The Art Bookshop, London UK</title>
  </head>
  <body>
    <div id="header">
      <h1>The Art Book Shop</h1>
      <ul>
        <li><a href="index.html">home</a></li>
        <li><a href="index.html">new publications</a>
            </li>
        <li class="current-page">
            <a href="index.html">contact</a></li>
      </ul>
    </div><!-- end header -->
    <div id="content">
      <p>Charing Cross Road, London, WC2, UK</p>
      <p><span class="contact">Telephone</span>
        0207 946 0946</p>
      <p><span class="contact">Email</span>
        <a href="mailto:books@example.com">
        books@example.com</a></p>
      <iframe width="425" height="275" frameborder="0"
        scrolling="no" marginheight="0" marginwidth="0"
        src="http://maps.google.co.uk/maps?f=q&amp;
        source=s_q&amp;hl=en&amp;geocode=&amp;
        q=charing+cross+road+london&amp;output=embed">
        </iframe>
    </div><!-- end content -->
    <p>&copy; The Art Bookshop</p>
  </body>
</html>
```

▸ `DOCTYPES` tell browsers which version of HTML you are using.

▸ You can add comments to your code between the `<!--` and `-->` markers.

▸ The `id` and `class` attributes allow you to identify particular elements.

▸ The `<div>` and `<span>` elements allow you to group block-level and inline elements together.

▸ `<iframes>` cut windows into your web pages through which other pages can be displayed.

▸ The `<meta>` tag allows you to supply all kinds of information about your web page.

▸ Escape characters are used to include special characters in your pages such as <, >, and ©.